

KAIST EureCar: 복잡한 도로 상황에서의 환경인식 및 충돌회피 가능한 무인자율주행차량 개발

KAIST 이웅희·윤상열·심인욱·신승학·최종원·오지원·심현철·권인소·최세범

1. 서론

무인자율주행차량은 오래전부터 관심의 대상 이었으나 전자기술의 발달로 소형컴퓨터 및 센서의 개발이 가능하게 된 1980년대부터 개발이 시작되어 소형 카메라, 레이저 센서, GPS등이 보급되기 시작한 1990년대에 본격적으로 개발 되었다.

무인자율주행기술의 비약적인 진보는 미국 DARPA가 2004년부터 주최한 Grand Challenge에서 사람의 도움없이 도로를 스스로 주행하는 기술 개발의 가능성을 열었으며[1] 2007년 Urban Challenge를 통하여 시가지에서의 무인자율 주행기술이 크게 발달되었다[2]. 최근에는 Stanford대학의 기술에 기반한 Google 무인자동차가 미국 네바다주와 캘리포니아 주에서 도로 자율 주행이 가능 하게끔 면허를 취득하는 수준까지 발전 하였다.

국내에서는 현대기아자동차에서 주최한 무인 자율주행경진대회가 2009년과 2011년 개최 되었다. 제1회 대회에서는 고정된 비교적 단순한 장애물을 회피하면서 경로를 완주하는 것이 요구되었으며, 제2회 대회서는 고정 및 이동식 장애물 회피, 복잡한 장애물 회피, 신호등 검출 및 보행자 검출 등의 복잡한 임무의 수행이 요구되었다.

본교에서는 2009년 제1회 대회때부터 기계공학과, 항공우주공학과, 전자공학과와의 연구진의 융합 연구를 통해 자율주행 차량을 개발하여 2012년 제2회 대회에서는 9개 미션중 8개를 모두 성공적으로 수행하는 성과를 거두었다.

이 논문에서는 본 연구진이 개발한 무인자율 주행시스템의 전반적인 기능 설명 및 시스템의 구성에 대해 설명하고 차량 제어를 위한 제어 시스템의 구성과 장애물과 영상 인지를 위한 시스템에 대하여 설명한다. 새로운 경로 표현법을 제안하고 이를 이용한 장애물 회피를 위한 경로 생성과 관리, 충돌 여부 확인 대한 방법을 제안하며 시뮬레이션과 실제 무인차량 대회를 포함한 여러 차례의 실험을 통하여 그 성능을 검증하였다.



[그림 1] KAIST EureCar - 무인자율주행차량

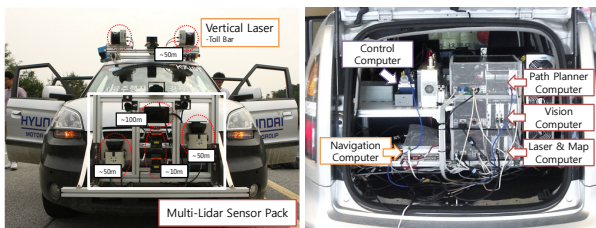
2. 무인자율주행시스템의 구성

무인자율주행 차량은 차량 주변의 환경을 인지하기 위한 시스템, 차량의 절대 위치를 파악하기 위한 시스템, 장애물을 회피하기 위한 경로와 속도를 생성하는 시스템과, 생성된 경로와 속도를 추종하기 위한 제어 시스템으로 구성된다. 주변 환경 인지 시스템을 구축하기 위해서는 카메라 혹은 레이저 스캐너와 같은 센서를 단

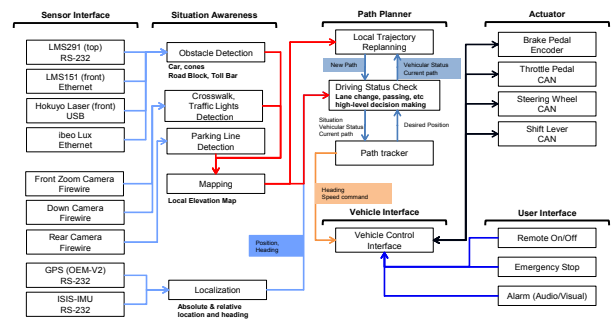
독으로 이용하는 것 보다는 다양한 센서를 복합적으로 활용하여 각 센서의 단점을 보완하는 것이 정확한 주변 환경 인지에 유리하다. 무인 자율주행 차량은 실시간성이 매우 중요하기 때문에, 차량에 장착되어 있는 센서들을 통해 얻은 데이터를 바탕으로 장애물과 충돌하지 않도록 실시간 경로를 생성하는 것이 중요하다.

KAIST EureCar팀에서 개발한 차량은 항법을 위한 2개의 DGPS와 IMU와 장애물 감지를 위한 8개의 Laser scanner 및 영상처리를 위한 카메라 4대가 장착되어 있으며, 차량의 주행 등을 제어하는 실시간 제어용 컴퓨터와 영상 및 레이저 데이터 및 경로 생성을 담당하는 3대의 컴퓨터가 설치되어 있다. 영상 정보를 처리하는 컴퓨터와 레이저 정보를 처리하는 컴퓨터는 다양한 외부환경의 정보를 수집하여 네트워크를 통해 경로 생성을 담당하는 컴퓨터로 처리된 정보를 전달하고, 경로 생성 컴퓨터는 이를 활용하여 최적의 경로를 생성한다. 각 컴퓨터는 정보 전달의 지연을 최소화하기 위해서 공유기나 외부 라우터를 통하지 않고 직접적으로 1:1 통신을 하도록 디자인하였다. 비전 컴퓨터는 이미지 정보를 수집하고 이를 이용하여 각 미션에 필요한 데이터 처리를 거친 후 결과를 수집한다. 그리고 수집된 각 미션별 정보들을 하나로 Packing하여 UDP방식으로 레이저 컴퓨터로 전송한다. 레이저 컴퓨터는 레이저 정보를 이용하여 지역적으로 장애물을 판별하기 위한 Elevation Map을 생성한다. 생성된 맵은 레이저를 이용한 정보뿐만 아니라, 비전 컴퓨터에서 처리된 정보를 포함하고 있다. 이렇게 생성된 맵은 경로생성 컴퓨터로 전송된다.

본 시스템의 최대 장점은 센서와 소프트웨어의 독립성이다. 시스템이 항상 안정적으로 유지하기 위해 이렇게 각 하드웨어 센서 모듈과 소프트웨어 모듈을 독립적



[그림 2] 영상 및 레이저 센서의 장착위치(좌) 무인 자율주행차량의 컴퓨팅 시스템(우)



[그림 3] 차량에 설치된 센서와 시스템 구조도

으로 구성하였다. 특히 주행 중에 카메라 센서나 레이저 센서가 부서져서 작동하지 못하는 경우가 발생해도 영향을 받는 미션을 수행하지 못할 뿐이지 전체 시스템은 계속 작동될 수 있도록 고안되었다.

2.1 차량 제어 시스템

차량제어시스템은 차량을 무인으로 구동하기 위해 차량에 장착한 액츄에이터 제어와 차량의 상태를 ECU로 부탁 받기위한 CAN 통신 및 액츄에이터의 상태를 확인위한 센서들로 구성되어있다. 경로 계획 컴퓨터로부터 받은 제어 입력을 페달 구동을 위한 모터 컨트롤용 PWM으로 바꾸고 feedback을 위해 엔코더로부터 캠의 회전각을 받아들이며 횡방향 제어를 위해 CAN통신을 통하여 핸들을 조작한다. 또한 전후진을 제어를 위하여 차량의 기어에 부착된 선형 모터를 제어한다. 추가적으로 차량의 비정상적인 주행에 대비하여 원격/직접 E-Stop 스위치와 연결 되어 있어 비상시 차량을 정지 시키는 기능을 수행한다.

2.2 항법 시스템

무인 자율 주행 차량이 주어진 경로를 따라 자율주행을 하기 위해서는 차량의 위치, 진행 방향 등의 기본 운동 변수를 실시간으로 추정할 수 있어야 한다. 이를 위한 적합한 센서로는 소형 무인 항공기에 많이 사용되는 strap-down GPS-aided INS(Inertial Navigation System, 관성항법장치)가 있는데 정밀도가 높아 무인 자율 주행 차량에 사용되기 적합하다. 관성항법장치는 차량에 고정된 3축 가속도계와 3축 각속도계로 구성된 관성측정장치 (Inertial Measurement Unit, IMU) 로부

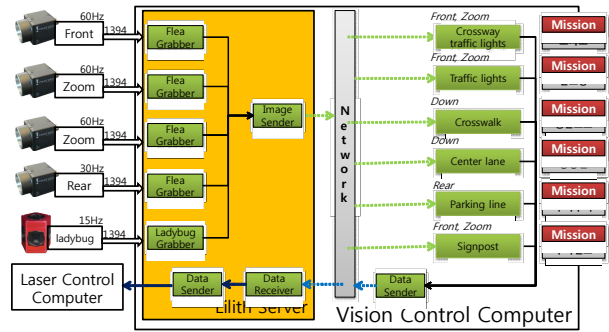
터 측정되는 차량의 가속도, 각속도를 적분하여 차량의 속도, 위치, 자세, 방향각을 계산하며, IMU의 bias와 drift에 의해 축적된 오차를 GPS 위치 정보를 이용하여 교정한다.

관성측정장치가 제공하는 가속도와 각속도는 시리얼 포트를 통해 항법 컴퓨터에 입력되며, 이는 100Hz로 실행되어 선속도, 위치, 자세각 및 방향을 계산하게 된다. 관성센서의 누적오차는 DGPS의 위치정보가 계산될 때마다 5Hz로 실행되는 Extended Kalman Filter에 보내져 오차가 추정 및 보정된다. 본 연구에 사용된 IMU와 GPS는 직렬통신으로 연결된다. 항법 컴퓨터와 항법 센서는 RS-232의 통신 규약을 따르고 항법 컴퓨터와 Path Planner PC 간의 통신은 Ethernet 통신 규약으로 항법 정보를 전달한다.

본 연구에 사용된 PC104기반 항법 시스템은 PC104기반 상업용 PC와 GPS, IMU 센서로 구성되어 있다.

2.3. 비전 시스템

[그림 4]는 보면 비전 처리 컴퓨터가 동작하는 전체 구조를 나타낸다. 본 시스템에서는 총 4개의 Flea 카메라와 1개의 레이더백 카메라를 사용한다. 모든 카메라는 IEEE 1394포트를 사용하여 비전 컴퓨터와 연결하였다. 각 카메라에서 영상 데이터를 자체 개발한 네트워크 서버인 Lilith 서버가 수집하여 네트워크 상에 UDP방식으로 영상을 전달한다. 이 때 이미지의 용량이 크면 이미지 전송 속도에 영향을 미칠 수 있으므로 수집된 영상을 압축하는데, 압축된 이미지의 크기에 따라 가변적으로 압축률을 결정하여 네트워크의 속도를 유지하였다. 이렇게 네트워크 상에 방영되는 데이터는 각 개별 알고리즘 모듈이 필요에 따라 네트워크에서 이미지를 추출하여 이용함으로써 알고리즘 모듈과 센서 모듈의 독립화를 구현하였다. 이렇게 센서 모듈을 독립적으로 구성함으로써 구조적 강인함을 확보하였다. 본 시스템에서는 주행 중 센서가 부서지거나 고장이 나도 문제없이 동작하는 것을 확인하였다. 또한 이렇게 분리된 모듈을 사용함으로써 각 모듈 별 알고리즘 처리 속도에 대한 동기화문제도 함께 해결하였다. 대회에서 사용된 비전 처리 모듈은 신호등, 교차로, 횡단보도, 주차구역,



[그림 4] 비전 컴퓨터 처리 구조도

주차번호를 확인한다. 각 모듈은 독립적으로 이미지를 처리하여 출력된 결과를 레이저 컴퓨터로 전송을 한다.

일반적으로 영상 개발 분야에서 다루는 인식 알고리즘들은 오프라인에서 이루어지는 경우가 많기 때문에 무인차와 같이 제한된 컴퓨팅 환경에서 실시간으로 구동 가능한 것이 많지 않다. 또한 최근 개발되고 있는 무인차 알고리즘의 경우 대상의 위치를 미리 저장해놓고 인식을 시도한다.[3] 이는 계산 시간이 적게 걸리고 인식 성능을 획기적으로 향상시킬 수 있지만, 저장되어 있지 않은 대상에 대해서는 전혀 탐지를 못하게 된다는 단점이 있다.

본 논문에서는 무인차와 같은 제한된 컴퓨팅 환경에서 미리 저장된 위치 정보 없이 교통 장애물들을 인지할 수 있는 실시간 알고리즘들을 제안한다.

2.3.1 신호등 및 횡단보도 인식 알고리즘

신호등 인식 알고리즘은 신호등의 색 정보와 형태정보를 이용한다. 외부 빛 변화에 강인한 HSV도메인을 이용하였으며 확률적 템플릿 매칭을 통해 신호등의 색과 형태에 가장 가까운 영역을 추출한다.

횡단보도 인식의 경우 1차원 평균필터를 이미지 전체에 적용한 이미지와 원본 이미지의 차 영상을 이용해 횡단보도의 패턴을 인식한다. 이 때 강인한 인식을 위해서 절대값을 이용한 차 영상을 사용하지 않고 음수 영역과 양수 영역에 해당하는 두 개의 차 영상을 얻어 각각 사용한다.

이 두 알고리즘은 다른 알고리즘들에 비해 오탐지율이 높기 때문에 신호등과 횡단보도는 함께 위치한다는 가정을 통해 알고리즘의 신뢰도를 높였다.

2.3.2 방향지시등 인식 알고리즘

이 알고리즘은 신호등 탐지와 방향 인식의 두가지로 나뉜다. 신호등 탐지의 경우 신호등 탐지에서 사용된 확률적 템플릿 매칭이 적용되었다. 그 이후 탐지된 영역의 좌우에 대해 각각 평균 이상의 밝기를 가진 픽셀들의 개수를 세어 화살표의 방향을 인식하였다.

2.3.3 차단봉 인식 알고리즘

차단봉을 인식하기 위해서 노란색과 검은색의 경계가 주기적으로 위치한 성질을 이용하였다. 각 색상 도메인에 가중치를 주는 1차원 미분 필터를 이용해 경계선을 찾은 뒤 랜덤으로 경계선에 해당하는 영역들을 지정한다. 이 영역들을 잇는 펄스형태 필터를 적용하여 주기적으로 반복되는지의 여부를 결정한다. 이 과정을 반복해 최종 결과를 출력한다.

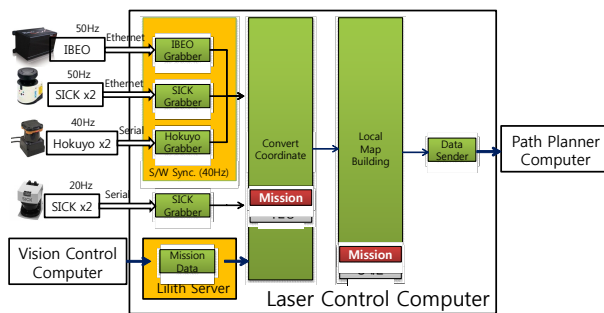
2.3.4 주차 구역 인식 알고리즘

주차구역의 경우 정확한 주차선의 위치를 파악해야 하므로 입력된 영상을 위에서 보는 형태로 워핑시킨다. 이후 워핑된 영상에서 Canny-edge를 추출하고 그 결과에서 RANSAC을 이용해 직선들을 탐지한다. 이후 주차선의 크기와 형태에 대한 가정들을 통해 오탐지된 선들을 삭제하고 정확한 주차선의 위치를 인식한다.

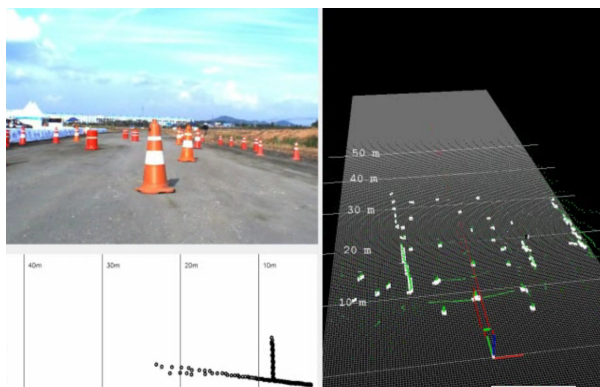
2.4 레이저 정보처리 시스템

[그림 5]은 레이저 정보처리 컴퓨터의 동작을 설명한다. 레이저 처리 컴퓨터에는 총 7개의 레이저 데이터를 처리하게 된다. IBEO는 100m 정도의 장거리 물체를 발견하기 위해서 설치되었고, SICK사의 LMS151은 50m 정도의 중거리 물체를 발견하기 위해서 설치하였으며, Hokuyo는 15m 정도의 단거리 물체를 측정하기 위해 설치되었다. 그리고 LMS291은 차단봉을 발견하기 위해 설치되었다 [그림 7].

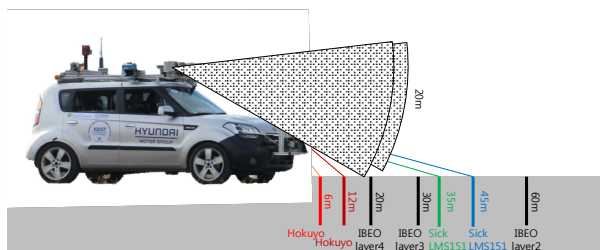
이를 위해서는 모든 레이저의 캘리브레이션이 필수적이며, 레이저간의 정확한 동기화도 반드시 필요하다. 이렇게 적응된 레이저만으로 장애물과 언덕 그리고 자동차가 갈 수 있는 길과 없는 길을 판단하였다. 장애물에 대한 판별은 점군을 직접 사용하지 않고 Elevation



[그림 5] 레이저 컴퓨터 처리 구조도



[그림 6] 영상(좌상) 수직 레이저 스캔 데이터(좌하) Local Elevation Map(우)



[그림 7] 레이저 스캐너 센싱 범위

Map을 사용하였다 [그림 6]. 각 Cell에 속하는 점군 데이터의 기울기 값으로 장애물을 판별하였다. 이렇게 기울기를 이용하여 장애물을 판별하면, 별도의 처리 없이 언덕과 같은 것이 장애물로 잘못 판단되는 일이 없어진다. 만약 이렇게 땅에 수평한 방향의 레이저를 적응하지 않고 사용한다면, 언덕 등과 같이 지나갈 수 있는 영역을 장애물로 판별하지 않기 위해 별도의 처리과정을 거쳐야 한다. SICK 291은 차량의 위에 옆으로 90도 기울어지도록 설치하였는데, 이는 전방에 존재하는

차단봉을 선행적으로 판별하기 위해 설치되었다. 레이저 컴퓨터에도 보조용 카메라가 설치되어 있는데, 이 카메라로는 차단봉이 매우 작게 보여 비전 정보로 차단봉을 사전에 판단하기가 매우 어려웠다. 바닥에 수직으로 설치된 레이저로 이 차단봉의 대략적인 위치를 판별하도록 하였고, 해당 위치에서 레이저 보조용 카메라로 차단봉의 패턴을 파악하여 차단봉을 검출하였다. 또한 레이저 컴퓨터는 기본적인 장애물을 판별하는 것 외에도 비전 컴퓨터에서 처리된 데이터를 Map에 추가시켜 비전에서 처리된 정보도 포함시켰으며, 비전 컴퓨터에서 카메라를 이용한 적당히 측정된 거리값을 레이저를 이용하여 정확한 거리를 측정하고, 잘못된 False Alarm을 제거시켜주는 역할도 하였다. 이렇게 처리된 데이터는 다시 최소한의 크기로 Encoding하여 경로 생성 컴퓨터로 전송된다.

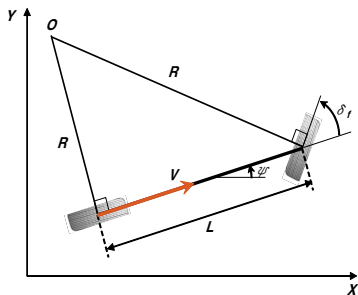
3. 차량 제어

3.1 차량 기구학 모델

차량 제어를 위한 기준 차량 기구학 모델을 차량의 뒷바퀴 중심에서 [그림 8]과 같이 모델링하였다. 차량의 바퀴와 지면에는 slip이 없다고 가정하였으며 좌우측 바퀴를 하나의 바퀴로 가정한 모델을 사용하였다. 차량의 기구학 모델과 회전 반경을 차량 뒷바퀴 중심축으로 유도하면 다음과 같다(식 1) [4,5].

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} V \cos(\psi) \\ V \sin(\psi) \end{bmatrix} \quad (1)$$

$$R = L / \tan(\delta_f)$$



[그림 8] 차량기구학모델

3.2 횡방향제어 알고리즘

무인 자율 로봇의 이동 경로 추종을 위한 항법 알고리즘인 경로점 기반 항법은 저속 주행을 하거나 방향 전환이 용이한 로봇에서 많이 사용되어 왔다. 경로점 기반 항법을 이용시 경로점이 전환 되면서 발생하는 불연속적인 제어 명령이 차량의 불안정한 운동을 야기하고 경로점을 통한 장애물 감지는 회피 경로 수정에 부적합한 표현 방식으로 본 연구에서는 다항식을 이용한 연속적인 경로를 정의 하고 기존의 경로점 추종 알고리즘을 이용하여 차량 제어를 하였다.

3.2.1 경로 정의

경로(Path)는 주어진 경로점을 일정 구역으로 나누어 n개의 3차 방정식으로 구성한다. 3차 방정식의 계수가 무한대로 정의 되는 것을 막기 위해 각각 경로의 시작점을 원점에 위치하고 기울기가 0 이 되도록 좌표 변환시켜 Curve Fitting을 수행한다. 위 조건으로 3차 방정식 저차 계수들은 0이 되며 식으로 표현하면 식(2)과 같다.

$$y_{path} = ax_{path}^3 + bx_{path}^2 \quad (2)$$

$$P_n = [x_{n,ENU}, y_{n,ENU}, h_{n,ENU}, a_n, b_n, L_n, V_n]^T \quad (3)$$

n 번째 경로에 대하여 (3)와 같이 배열로 간단히 표현할 수 있으며 경로점으로 정의 하는 것에 비하여 배열의 크기를 크게 감소 시킬 수 있다. [그림 9]의 왼쪽 그림에는 50여개 경로점을 오른쪽의 10여개의 경로 표현방식을 통해 더 적은 수의 배열로 표현할 수 있다.

경로 P_n 에서 $x_{n,ENU}$, $y_{n,ENU}$ 와 $h_{n,ENU}$, $y_{n,ENU}^{ENU}$ 는 ENU 좌표계에서의 각각 경로의 시작 위치와 방위각을 나타내고 [그림 10], L_n 는 경로의 길이를 V_n 은 그 경로의 최대 속도(제한속도)를 나타낸다.]

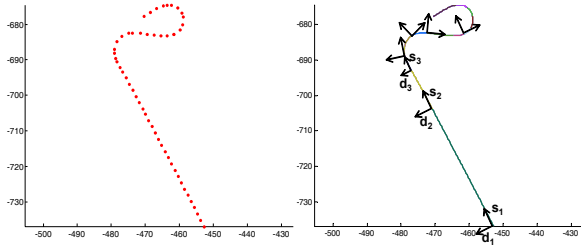
3.2.2 경로 추종

경로를 추종하기 위해 경로 위의 한 점을 제어점으로 잡는다. 제어점은 차량의 진행 방향에 앞쪽에 차량 속

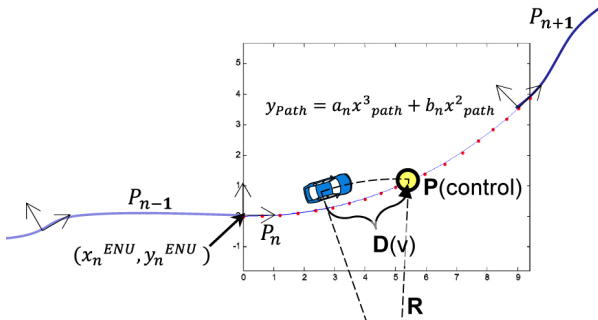
도에 맞춰 거리, $D(I)$ 가 가변적으로 바뀌는 위치에 지정 된다[그림 10]. 경로 위의 제어점 X_{path} 는 좌표변환을 통해 차량의 고정좌표계로 변환 시킬 수 있으며 고정좌표계에서의 차량의 회전 반경과 제어점간의 관계는 식 (4)와 같다.

$$\frac{1}{R} = \frac{-2(y_{p,body})}{(x_{p,body})^2 + (y_{p,body})^2} \quad (4)$$

차량이 제어점에 도달하기 위한 조향각 δ_r 는 식 (1)과 (4)를 통하여 구할 수 있다.



[그림 9] 경로점(좌)과 경로(우)

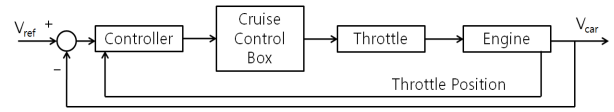


[그림 10] 경로와 제어점

3.3 종방향제어 알고리즘

3.3.1 Throttle pedal 제어 알고리즘

가속 페달 제어를 위한 기본적인 블록도는 [그림 11]와 같이 그려진다. 현재 차량의 종방향 속도는 차량 CAN 통신을 통해 입력 받는다. 측정되는 종방향 속도 정보는 제어 목표인 V_{ref} 와 함께 제어기로 입력된다. 현재 속도가 목표 제어 속도를 추종하는 것이 가속 페

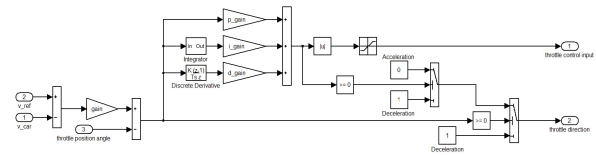


[그림 11] 종방향 제어를 위한 가속 페달 제어 블록 다이어그램

달 제어의 목표이며, 이를 위한 제어 입력이 앞서 설명한 cruise controller의 모터 입력이 된다. 이를 통해 차량의 가속 페달을 제어하고, 차량의 가속 및 감속 제어로 이어지게 된다.

차량의 속도와 Throttle position 정보를 CAN 통신을 통해 입력 받아, Cascaded Control Loop을 생성함으로써, 정상상태에서의 속도 오차를 효과적으로 감소시켰다.

[그림 12]은 간략화 된 가속페달의 제어 로직이다. CAN 정보를 통해 받는 차량 속도만으로 피드백 제어를 하기엔, 페달의 움직임과 차량 속도의 변화 사이에 큰 딜레이가 존재 하기 때문에 어려움이 있다. 그러나, 페달의 움직임과 스로틀 포지션 각도 사이에는 거의 딜레이가 존재하지 않는다. 그러므로, 제어기는 차량 속도 정보와 throttle position 정보를 동시에 받아 PID제어를 하는 방식으로 설계 되었다.



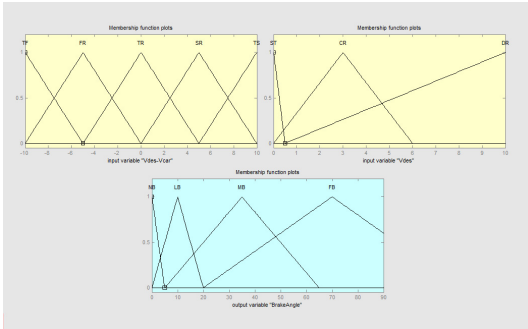
[그림 12] 가속 페달 제어를 위한 제어 로직

3.3.2 Brake pedal 제어 알고리즘

브레이크 페달 제어를 위한 기본적인 원리는 두 부분으로 나뉜다. 첫째로는 원하는 브레이크 페달의 밟힘 정도를 생성해 주는 desired brake angle generation 이며, 둘째로는 이렇게 생성된 desired brake angle 값을 실제로 모터 제어를 통하여 추종하는 추종 제어기이다. 이에 상응하여 캠은 브레이크 페달을 움직이게 되고, 그것은 곧 차량의 감속으로 이어진다. Desired brake angle 생성기 부분에서는 현재 차량 속도와 원하는 차량 속도를 입력으로 하는 fuzzy logic 을 통하여 원하

는 brake angle 값을 생성하게 되고, brake angle 추종 제어기는 desired brake angle 값과 엔코더가 측정한 현재 brake angle 값을 입력으로 하여 adaptive controller 를 사용해 모터를 제어 한다.

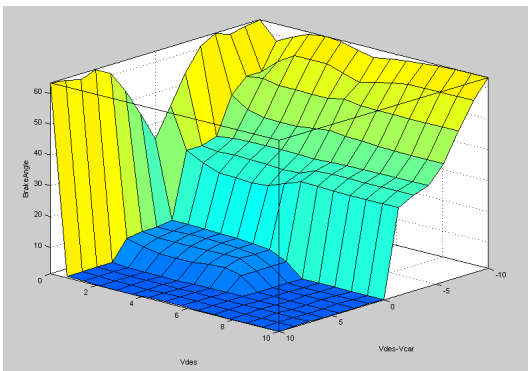
[그림 13]부터 [그림 15]은 desired brake angle 생성



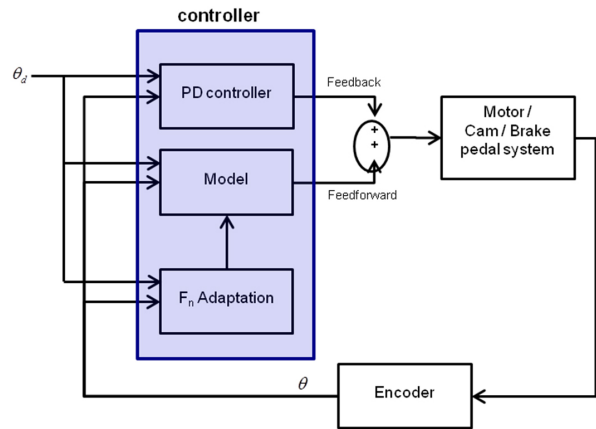
[그림 13] Desired brake angle 생성을 위한 fuzzy logic 의 membership functions

		V_{ref}		
		ST 0	CR 3	DR 10
$V_{ref} - V_{-3r}$	TS 10	FB	NB	NB
	SR 5	FB	NB	NB
	TR 0	MB	LB	NB
	FR -5	FB	MB	MB
	TF -10	FB	FB	FB

[그림 14] Desired brake angle 생성을 위한 fuzzy logic 의 rules



[그림 15] Desired brake angle 생성을 위한 fuzzy logic의 fuzzy surface



[그림 16] 모터 입력 결정을 위한 제어로직의 블록선도

을 위한 fuzzy logic을 설명하는 그림이다. 입력된 목표 차량 속도와 현재 차량 속도는 [그림 13]와 같은 membership functions 와 [그림 14]와 같은 fuzzy rule 들을 통하여 desired brake angle을 생성하게 된다.

이렇게 생성된 desired brake angle은 추종 제어기에 입력되어 모터 제어 입력을 결정한다. [그림 16]은 모터 제어를 표현한 블록선도이다.

4. 경로 계획

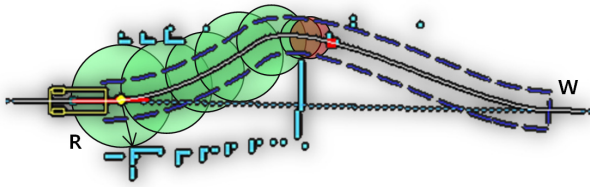
경로 계획은 경로 상의 장애물 검출과 장애물 회피를 위한 경로 생성으로 나뉜다. 장애물로 확정된 데이터 즉 Local Elevation Map과 주행 경로의 교차 여부를 판단한다. 경로 생성은 경로를 가로 막는 장애물을 회피하도록 기존 경로를 수정하여 새로운 대체 경로를 만든다.

4.1 장애물 검출

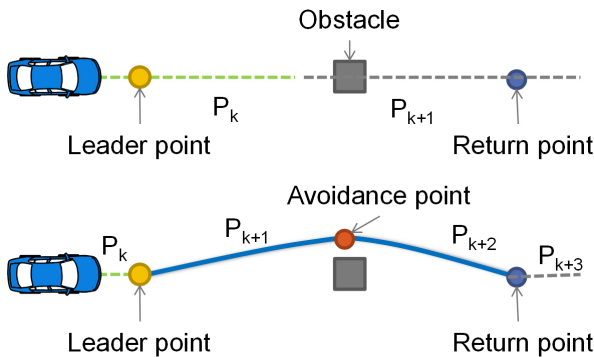
장애물 검출 알고리즘은 차량에서 [그림 10]의 제어점까지 검출과 제어점부터 센싱 범위까지 검출 알고리즘으로 나뉜다. 차량에서 제어점까지의 장애물 검출은 차량의 감속 및 비상 정지를 위한 속도 프로파일용이며, 차량이 회전반경의 따라 이동할 경우 생성되는 궤적에서 차량으로부터 제어점 까지 거리에 장애물 유무를 검출한다.

제어점부터 센싱 범위까지 장애물 검출 알고리즘은 차량이 경로를 잘 추종한다는 가정하에 차량 궤적 안에

서의 장애물과의 충돌 여부를 판단하면 되는데 [그림 17]와 같이 계산량과 계산시간을 줄이기 위해 경로를 따라 경로상에 한 점에서 가장 가까운 장애물 데이터까지의 거리를 반지름으로 갖는 원(CCSD)을 센서가 읽히는 범위 까지 오버랩 시켜 가면서 장애물의 충돌여부를 판별한다. 원의 크기가 차량의 폭(W) 보다 작아지는 위치가 장애물의 위치이고 충돌이 있다고 판단되면 장애물 회피 경로 생성 알고리즘을 수행하여 장애물을 회피한다.



[그림 17] Continuous Collision Search Disc



[그림 18] 장애물 회피 경로 생성과 관리

4.2 3차 다항식을 이용한 경로 생성

3차 다항식 식 (2)을 이용하여 경로를 생성하기 위해 2지점의 위치와 기울기가 필요하다. 장애물 검출 알고리즘에서 검출된 장애물을 회피하는 알고리즘은 제어점(Control point)으로부터 회피점(Avoidance point)으로 장애물을 회피하는 경로와 회피점으로부터 원래의 경로로 돌아오기 위한 복귀점(Return point)을 정하여 기존 경로로 복귀하는 경로를 생성한다. 제어점과 복귀점은 기존 경로상의 한 점이며 회피점은 장애물을 회피하기 위한 임의의 한 점이다[그림 18]. 회피점을 구하기 위하여 Voronoi cell과 VHF(Vector Field Histogram)을

이용하였으며 위의 알고리즘을 통해 얻어진 회피점들로부터 3차 방정식을 이용하여 회피 경로를 생성한 후 장애물 검출 알고리즘으로 생성된 회피경로의 충돌여부를 확인하고 충돌이 없는 경로를 택하여 최종적으로 경로를 수정한다. 회피 경로가 모두 막힌 경우 경로를 따라 후진하면서 경로를 탐색하여 새로운 경로를 찾는다.

4.2.1 Single Voronoi Cell을 이용한 경로 생성

기존의 Voronoi diagram은 유한개의 장애물 데이터로부터 각각의 장애물 데이터 중 하나씩만을 포함하는 영역으로 분할할 때 분할된 영역에서 이웃하는 영역까지의 가장 가깝게 나누는 것이며, 이를 이용한 경로 계획은 Voronoi edges로부터 Graph search 알고리즘을 통해 최적화된 경로를 찾는 것이다. 기존의 Voronoi diagram을 이용한 경로 계획은 Voronoi diagram과 최적의 경로를 찾는 알고리즘의 계산시간으로 인하여 실시간으로 사용하기 어려워 알고 있는 지형 정보에서 최적의 경로를 찾기에 효과적이며 실제 Laser scanner로부터 얻은 Point cloud에 적용하여 국지적 경로를 찾기에는 부적합하다. Voronoi diagram의 기본 개념을 이용하여 센서 범위내에서 실시간 경로계획에 사용하기 위해 다음과 같은 알고리즘을 고려 하였다.

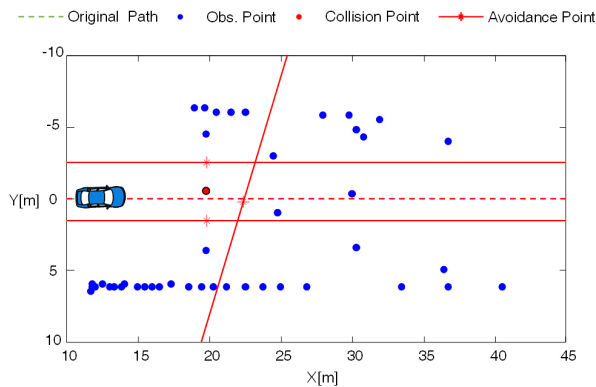
```
function SingleVoronoiCell
collision point(Obstacle points);
set rest points to Obstacle points
While (#(rest points) ≠ 0) {
    nearest point(collision point)
    perpendicular bisector(nearest point)
    set rest points to not near divided point set
    (rest points, bisector, collision point)
}
return perpendicular bisector set
```

기존의 경로를 가로막는 장애물 포인트에 대하여 Single Voronoi Cell 알고리즘을 통해 n개 Avoidance point 후보군을 구할 수 있다. Single Voronoi Cell 알고리즘은 일반적인 Voronoi Diagram과 달리 한 개의

Voronoi cell 만을 구하기 때문에 속도가 빨라 실시간으로 사용할 수 있다는 장점이 있으며 Voronoi Diagram의 정의에 의해 장애물과의 최대 clearance를 구하기 때문에 안정적으로 장애물을 피할 수 있다[그림 19]. 연속적으로 차량의 주행면서 Single Voronoi Cell 알고리즘을 통해[그림 20]과 같이 장애물을 회피하는 경로를 만들 수 있다.

4.2.2 Vector Field Histogram을 이용한 경로 생성

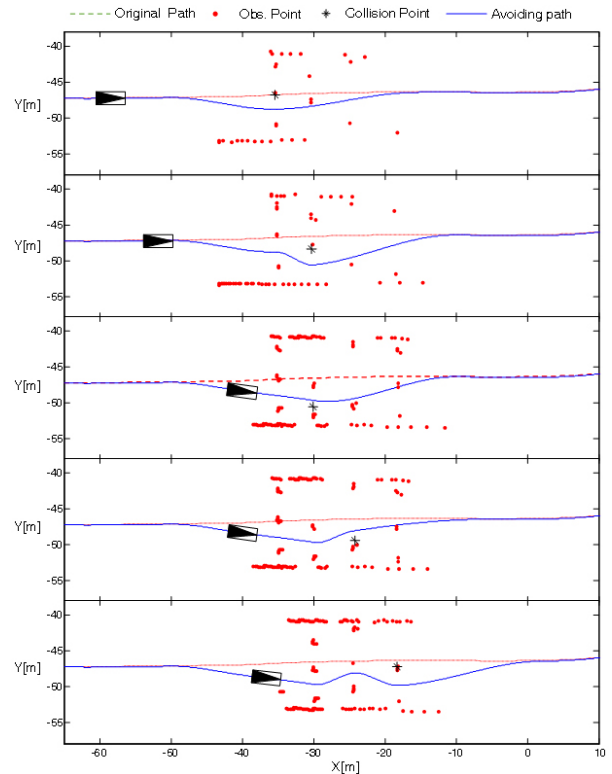
Vector Field Histogram은 Voronoi diagram이 장애물 간의 거리가 멀 때, 크기 우회 하는 경로를 생성하는 경우와 장애물이 주변에 없는 경우 경로를 생성하지 못하는 한계를 극복하기 위하여 사용하였다. VFH은 일반적인 VFH 알고리즘을 3차 방정식의 경로로 구성하기 위해 진행방향에 차량 속도에 비례하는 위치에 임의의 Avoidance point를 지정하는 알고리즘으로 수정하여 사용하였다.



[그림 19] Single Voronoi Cell을 이용한 경로 계획

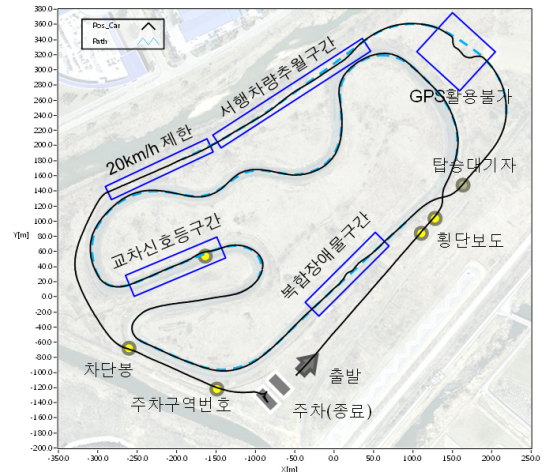
5. 주행 결과

[그림 21]은 현대NGV 무인자율주행 경진대회에서의 주행 결과이다. 총 9가지의 미션으로 구성되었다. 일시 정지하였다가 재출발하는 미션은 횡단보도 20km/h 제한구간내에서의 돌발장애물 발견시, 차단봉이 내려온 경우이다. 영상 관련 미션은 4가지로 교차신호등, 횡단보도, 탑승대기자, 주차등이 있고 경로를 생성하는 미션

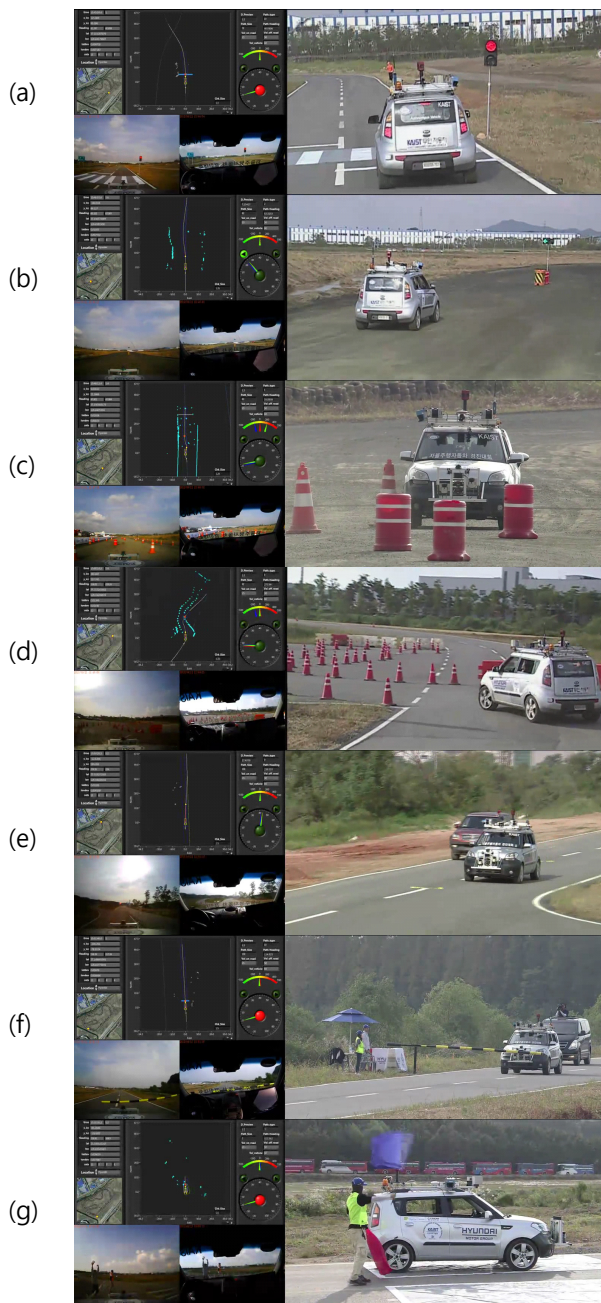


[그림 20] Single Voronoi Cell 알고리즘을 이용한 장애물 회피

은 교차신호등구간, 복합장애물구간, GPS 활용불가구간, 서행차량추월구간등이 있다. [그림 22]에서 각 미션의 구성과 경로계획 GUI를 확인할 수 있으며 [그림 23]에서와 같이 장애물들을 제안한 알고리즘을 통해 회피하는 결과를 볼 수 있다.이번 무인자율주행 경진대회

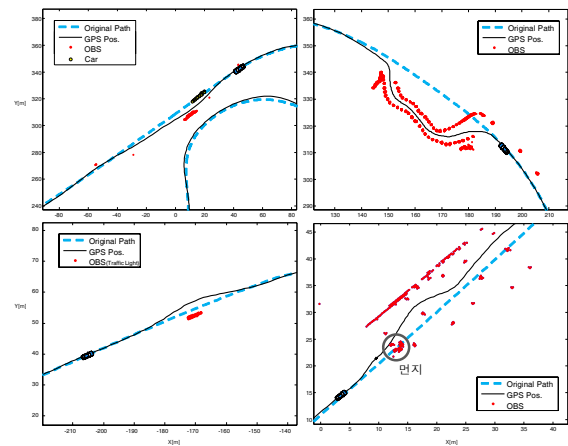


[그림 21] 현대NGV 무인자율주행 경진대회 주행 결과



[그림 22] (a) 횡단보도 (b) 교차신호등 (c) 복합장애물 (d) GPS활용불가 (e) 서행차량추월 (f) 차단봉 (g) 주차

에서 총 9개의 미션중 탑승대기자 검출을 제외한 횡단보도, 교차신호등, 복합장애물, GPS활용불가, 서행차량추월, 차단봉, 후진주차 모두를 성공적으로 수행하였다.



[그림 23] 서행차량회피(좌상), GPS활용불가(우상) 교차신호등(좌하), 복합장애물(우하)

6. 결론

본 논문에서는 카이스트 연구진이 개발한 무인자율주행차량의 구조와 영상 및 레이저 센서를 이용한 장애물 감지, 관성항법장치와 GPS를 기반한 항법 시스템 및 실시간 정밀차량제어 시스템을 소개하였다. 개발된 차량은 영상센서를 통한 신호등이나 보행자 검출이 가능하며, 다수의 레이저 센서를 사용하여 장애물을 감지하고 이들 정보를 실시간으로 처리하여 안전한 경로를 생성, 추종이 가능함을 보였다. 본 연구진은 현재까지 개발된 무인차량기술에 기반하여 보다 진보된 무인차량을 개발, 실제 도로에서 주변 차량 및 보행자를 감지하고 교통신호를 준수하면서 안전하게 주행하는데 필요한 제반기술을 개발하는데 계속 노력할 것이다.

후 기

본 논문은 현대기아자동차 (NGV) 무인자율주행경진대회의 지원 및 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임.(No. 20110031920)

참고문헌

- [1] Thrun, S., Montemerlo, M., et al, "Stanley: The robot that won the Darpa grand challenge,"

Journal of Field Robotics, 23(9), 661-692, 2006.

- [2] Urmson, C., Anhalt, J. et al, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, 25(8), 425-466, 2008.
- [3] Levinson, J., Askeland, J., Dolson, J., and Thrun, S., "Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles," *IEEE Conf. on International Conference on Robotics and Automation*, pages 5784-5791, 2011.
- [4] JAZAR, R. N. *Vehicle dynamics: theory and applications*, Springer, 2008.
- [5] RAJAMANI, R., *Vehicle dynamics and control*, Springer Science, 2006.



이 웅 희

2009 한국항공대학교 항공우주공학과(공학사)
2011 KAIST 항공우주공학과 (공학사)
2011~현재 KAIST 항공우주공학과 박사과정
E-mail : lamer0712@kaist.ac.kr



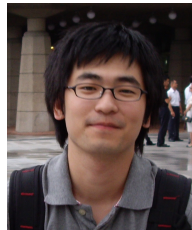
윤 상 열

1999 홍익대학교 기계공학과(공학사)
2001 GIST 기전공학 (공학사)
2009~현재 KAIST 항공우주공학과 박사과정
E-mail : sangyol.yoon@kaist.ac.kr



심 인 욱

2009 한양대학교 전산학과(공학사)
2011 KAIST 로봇공학학제 (공학사)
2011~현재 KAIST 미래자동차학제 박사과정
E-mail : iwshim@rcv.kaist.ac.kr



신 승 학

2009 연세대학교 전기전자공학(공학사)
2011 KAIST 로봇공학학제 (공학사)
2011~현재 KAIST 전기전자공학 박사과정
E-mail : shshin@rcv.kaist.ac.kr



최 종 원

2012 KAIST 전기및전자공학과 (공학사)
2012~현재 KAIST 전기및전자공학과 석사과정
E-mail : jwchoi@rcv.kaist.ac.kr



권 인 소

1981 서울대학교 기계설계학과 (공학사)
1983 서울대학교 기계설계학과 (공학사)
1990 Carnegie Mellon University Robotics
E-mail : iskweon@ee.kaist.ac.kr



오 지 원

2010 KAIST 기계공학과 (공학사)
2012 KAIST 기계공학과 (공학사)
2012~현재 KAIST 기계공학과 박사과정
E-mail : curoaura@kaist.ac.kr



최 세 범

1985 서울대학교 기계설계학과 (공학사)
1987 KAIST 기계공학과 (공학사)
1993 University of California, Berkeley Mechanical Engineering, (공학박사)
E-mail : sbchoi@kaist.ac.kr



심 현 철

1991 서울대학교 기계설계학과 (공학사)
1993 서울대학교 기계설계학과 (공학사)
2000 University of California, Berkeley Mechanical Engineering (공학박사)
E-mail : hcshim@kaist.ac.kr